

BÀI 3

FLUTTER VÀ CÁC WIDGET CƠ BẢN

GV: ThS. BÙI PHÚ KHUYÊN



* Cập nhật: 15.07.2024

NỘI DUNG BÀI HỌC

- Giới thiệu tổng quan về Flutter
- Cấu trúc dự án Flutter
- Widgets
- Layout
- Asset & Media
- Navigation & Routing



☐ Flutter là gì?

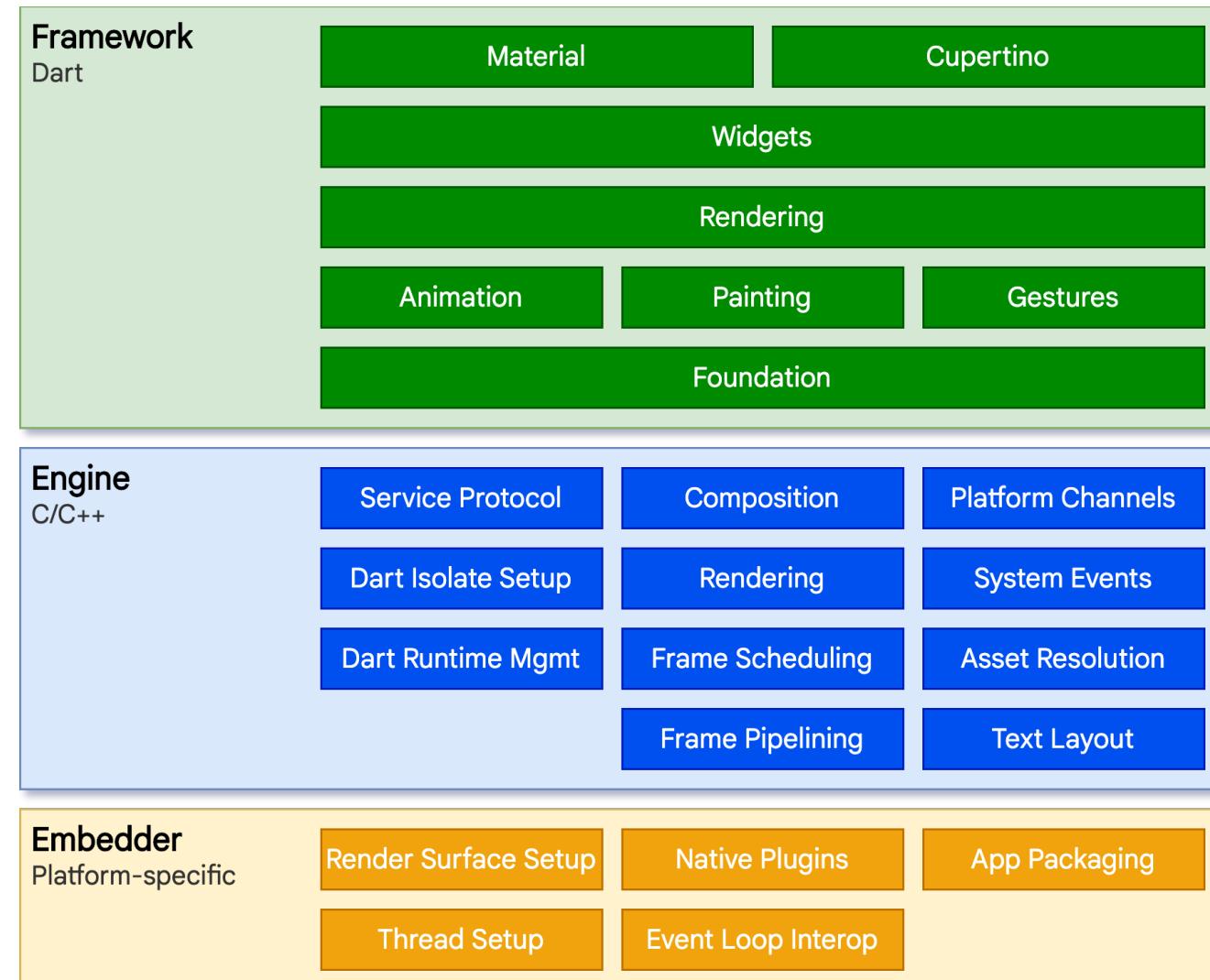
- Flutter là bộ công cụ UI mã nguồn mở do Google phát triển.
- Dùng để xây dựng các ứng dụng đa nền tảng từ một mã nguồn duy nhất: iOS, Android, Web, Desktop.

☐ Ngôn ngữ lập trình: Sử dụng Dart



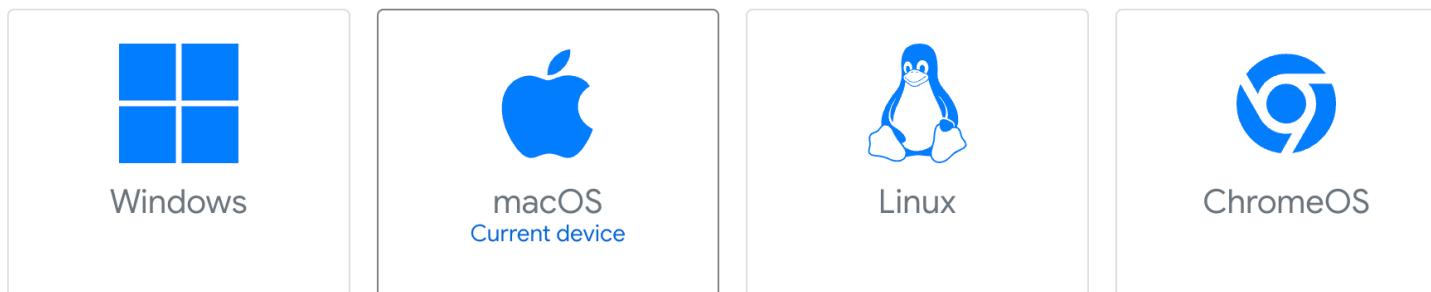
☐ Điểm mạnh

- **Hot Reload:** Xem thay đổi giao diện ngay lập tức mà không cần biên dịch lại toàn bộ.
- **Hiệu suất cao:** Ứng dụng được biên dịch thành mã máy (x64 và ARM).



○ Bước 1: Cài đặt Flutter SDK từ docs.flutter.dev.

Get started > Install



⚠ Important

If you develop apps in China, check out [using Flutter in China](#).

Unless stated otherwise, the documentation on this site reflects the latest stable version of Flutter. Page last updated on 2024-07-07. [View source](#) or [report an issue](#).

Chọn hệ điều hành phát triển

Download then install Flutter

To install Flutter, download the Flutter SDK bundle from its archive, move the bundle to where you want it stored, then extract the SDK.

1. Download the following installation bundle to get the latest stable release of the Flutter SDK.

[flutter_windows_3.24.2-stable.zip](#)

For other release channels, and older builds, check out the [SDK archive](#).

The Flutter SDK should download to the Windows default download directory: `%USERPROFILE%\Downloads`.

If you changed the location of the Downloads directory, replace this path with that path. To find your Downloads directory location, check out this [Microsoft Community post](#).

Tải Flutter SDK Windows

Download then install Flutter

To install Flutter, download the Flutter SDK bundle from its archive, move the bundle to where you want it stored, then extract the SDK.

1. Download the following installation bundle to get the latest stable release of the Flutter SDK.

Intel Processor

[flutter_macos_3.24.2-stable.zip](#)

Apple Silicon

[flutter_macos_arm64_3.24.2-stable.zip](#)

For other release channels, and older builds, check out the [SDK archive](#).

Tải Flutter SDK MacOS

○ Bước 2: Thêm Flutter SDK vào PATH.

Update your Windows PATH variable

To run Flutter commands in PowerShell, add Flutter to the [PATH](#) environment variable. This section presumes that you installed the Flutter SDK in `%USERPROFILE%\dev\flutter`.

1. Press `Windows + Pause`.
If your keyboard lacks a `Pause` key, try `Windows + Fn + B`.
2. Click [Advanced System Settings > Advanced > Environment Variables...](#)

The [Environment Variables](#) dialog displays.

3. In the [User variables for \(username\)](#) section, look for the [Path](#) entry.

1. If the entry exists, double-click on it.

The [Edit Environment Variable](#) dialog displays.

1. Double-click in an empty row.
2. Type `%USERPROFILE%\dev\flutter\bin`.
3. Click the `%USERPROFILE%\dev\flutter\bin` entry.
4. Click [Move Up](#) until the Flutter entry sits at the top of the list.
5. Click [OK](#) three times.

2. If the entry doesn't exist, click [New....](#)

The [Edit Environment Variable](#) dialog displays.

1. In the [Variable Name](#) box, type `Path`.
2. In the [Variable Value](#) box, type `%USERPROFILE%\dev\flutter\bin`
3. Click [OK](#) three times.

4. To enable these changes, close and reopen any existing command prompts and PowerShell instances.

Windows

Add Flutter to your PATH

MacOS

To run Flutter commands in the Terminal, add Flutter to the [PATH](#) environment variable. This guide presumes your [Mac runs the latest default shell](#), zsh. Zsh uses the `.zshenv` file for [environment variables](#).

1. Launch your preferred text editor.
2. If it exists, open the Zsh environmental variable file `~/.zshenv` in your text editor. If it doesn't, create `~/.zshenv`.
3. Copy the following line and paste it at the end of your `~/.zshenv` file.

```
export PATH=$HOME/development/flutter/bin:$PATH
```

bash

4. Save your `~/.zshenv` file.

5. To apply this change, restart all open terminal sessions.

If you use another shell, check out [this tutorial on setting your PATH](#).

Thiết lập biến môi trường cho Flutter SDK

○ Bước 3: Kiểm tra môi trường với lệnh flutter doctor.

```
Windows PowerShell x + v Windows
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\buiphukhuyen> flutter doctor

A new version of Flutter is available!
To update to the latest version, run "flutter upgrade".

Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.24.1, on Microsoft Windows [Version 10.0.22631.4037], locale en-US)
[✓] Windows Version (Installed version of Windows is version 10 or higher)
[X] Android toolchain - develop for Android devices
  X Unable to locate Android SDK.
    Install Android Studio from: https://developer.android.com/studio/index.html
    On first launch it will assist you in installing the Android SDK components.
    (or visit https://flutter.dev/to/windows-android-setup for detailed instructions).
    If the Android SDK has been installed to a custom location, please use
    'flutter config --android-sdk' to update to that location.

[X] Chrome - develop for the web (Cannot find Chrome executable at .\Google\Chrome\Application\chrome.exe)
  ! Cannot find Chrome. Try setting CHROME_EXECUTABLE to a Chrome executable.
[!] Visual Studio - develop Windows apps (Visual Studio Community 2022 17.11.1)
  X Visual Studio is missing necessary components. Please re-run the Visual Studio installer for the
"Desktop"
  development with C++" workload, and include these components:
    MSVC v142 - VS 2019 C++ x64/x86 build tools
    - If there are multiple build tool versions available, install the latest
      C++ CMake tools for Windows
      Windows 10 SDK
[✓] Android Studio (version 2024.1)
[✓] Connected device (2 available)
[✓] Network resources

! Doctor found issues in 3 categories.
```

```
buiphukhuyen — zsh — 95x32 MacOS
Last login: Wed Sep 11 09:01:47 on console
buiphukhuyen@Phu-Khuyens-MacBook-Pro ~ % flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.24.2, on macOS 14.6.1 23G93 darwin-arm64, locale en-VN)
[✓] Android toolchain - develop for Android devices (Android SDK version 35.0.0)
[✓] Xcode - develop for iOS and macOS (Xcode 15.4)
[✓] Chrome - develop for the web
[✓] Android Studio (version 2024.1)
[✓] IntelliJ IDEA Ultimate Edition (version 2024.2.0.2)
[✓] VS Code (version 1.92.2)
[✓] Connected device (3 available)
  ! Error: Browsing on the local area network for Phu Khuyen's iPhone. Ensure the device is
    unlocked and attached with a cable or associated with the same local area network as this
    Mac.
    The device must be opted into Developer Mode to connect wirelessly. (code -27)
[✓] Network resources

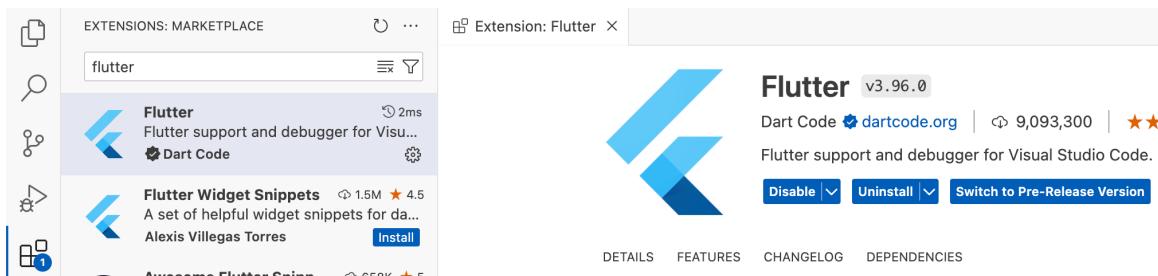
• No issues found!
buiphukhuyen@Phu-Khuyens-MacBook-Pro ~ %
```

THIẾT LẬP MÔI TRƯỜNG PHÁT TRIỂN {4}

- Bước 4: Cài đặt IDE (VSCode, Android Studio, XCode) và Flugin Flutter.

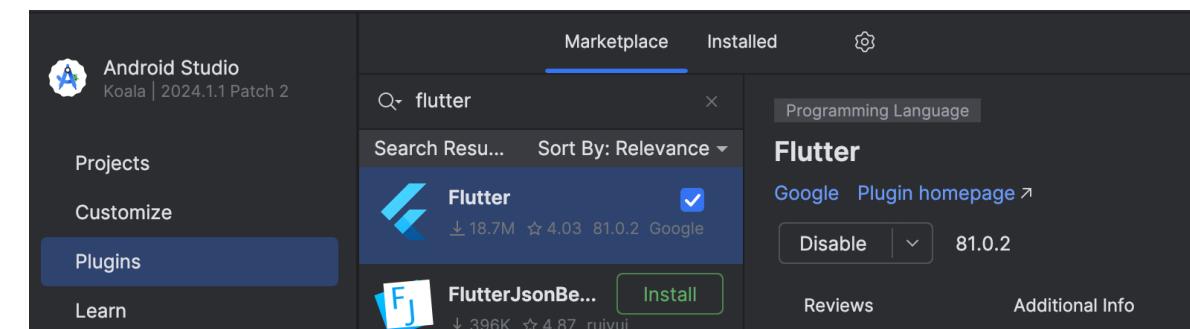


Visual Studio Code



The screenshot shows the Visual Studio Code Marketplace interface. A search bar at the top has "flutter" typed into it. Below the search bar, a list of extensions is displayed. The first item is "Flutter" by Dart Code, version v3.96.0, which is highlighted with a blue border. It has a rating of 4.5 stars and over 9 million installations. The description says "Flutter support and debugger for Visual Studio Code". Below the main list are two more items: "Flutter Widget Snippets" and "Awesome Flutter Snippets".

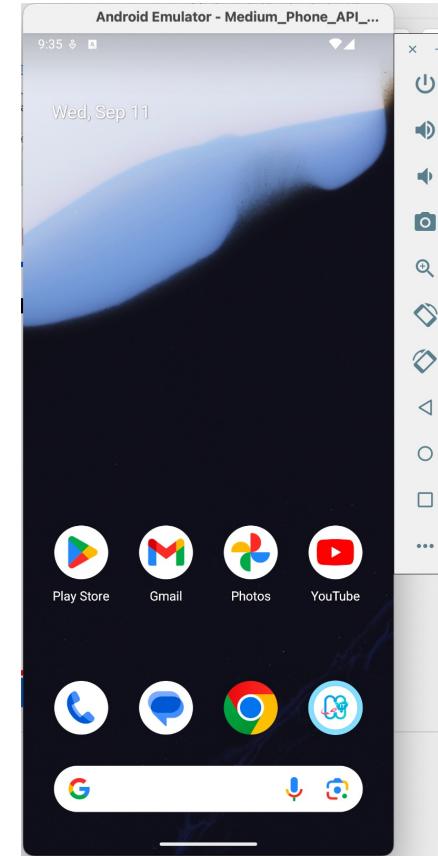
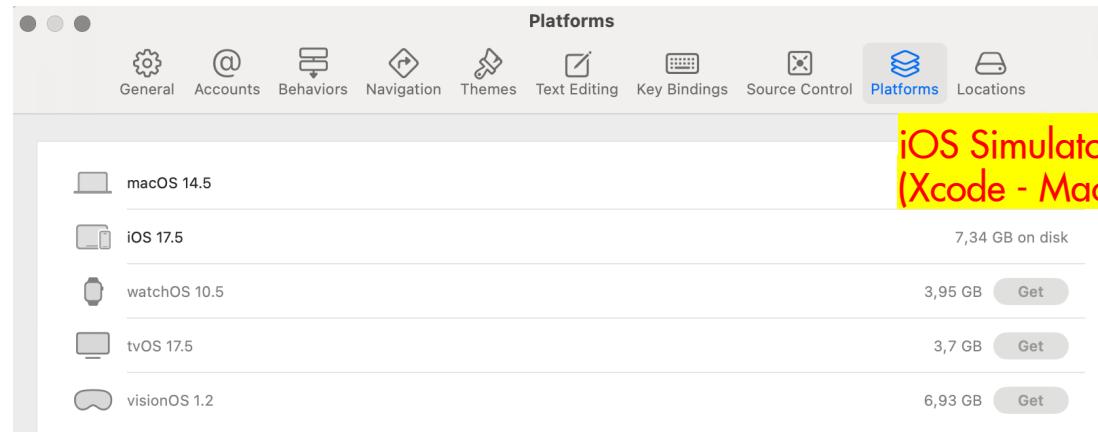
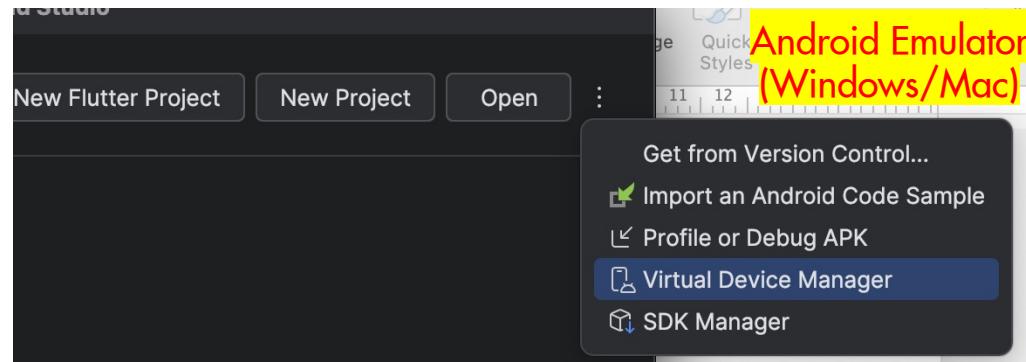
android studio



The screenshot shows the Android Studio Plugins Marketplace. A search bar at the top has "flutter" typed into it. Below the search bar, a list of plugins is displayed. The first item is "Flutter" by Google, version 81.0.2, which is highlighted with a blue border. It has a rating of 4.03 and over 18 million installations. The description says "Flutter support and debugger for Visual Studio Code". Below the main list are two more items: "FlutterJsonBe..." and "Reviews".

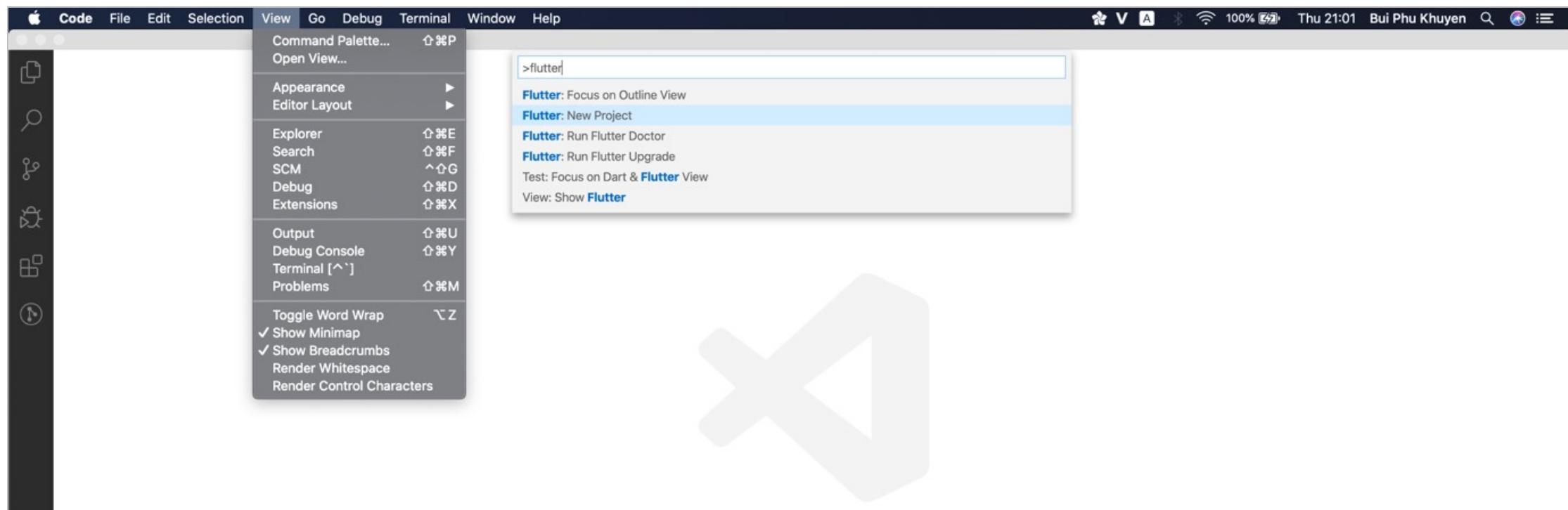
THIẾT LẬP MÔI TRƯỜNG PHÁT TRIỂN {5}

- Bước 5: Thiết lập thiết bị ảo (Android Emulator, iOS Simulator).



☐ Tạo mới dự án

- Sử dụng VSCode hoặc Android Studio.
- Hoặc lệnh: flutter create myflutterapp.



☐ Kiểm tra trạng thái dự án

```
● ● ● buiphukhuyen --zsh-- 95x32
Last login: Wed Sep 11 09:01:47 on console
buiphukhuyen@Phu-Khuyens-MacBook-Pro ~ % flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.24.2, on macOS 14.6.1 23G93 darwin-arm64, locale en-VN)
[✓] Android toolchain - develop for Android devices (Android SDK version 35.0.0)
[✓] Xcode - develop for iOS and macOS (Xcode 15.4)
[✓] Chrome - develop for the web
[✓] Android Studio (version 2024.1)
[✓] IntelliJ IDEA Ultimate Edition (version 2024.2.0.2)
[✓] VS Code (version 1.92.2)
[✓] Connected device (3 available)
```

```
OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS Flutter
[✓] VS Code: is fully installed. (version 1.42.1)
[✓] Connected device: is fully installed. (1 available)

In order to run your application, type:

$ cd .
$ flutter run

Your application code is in ./lib/main.dart.

exit code 0
```

Thông tin thiết bị

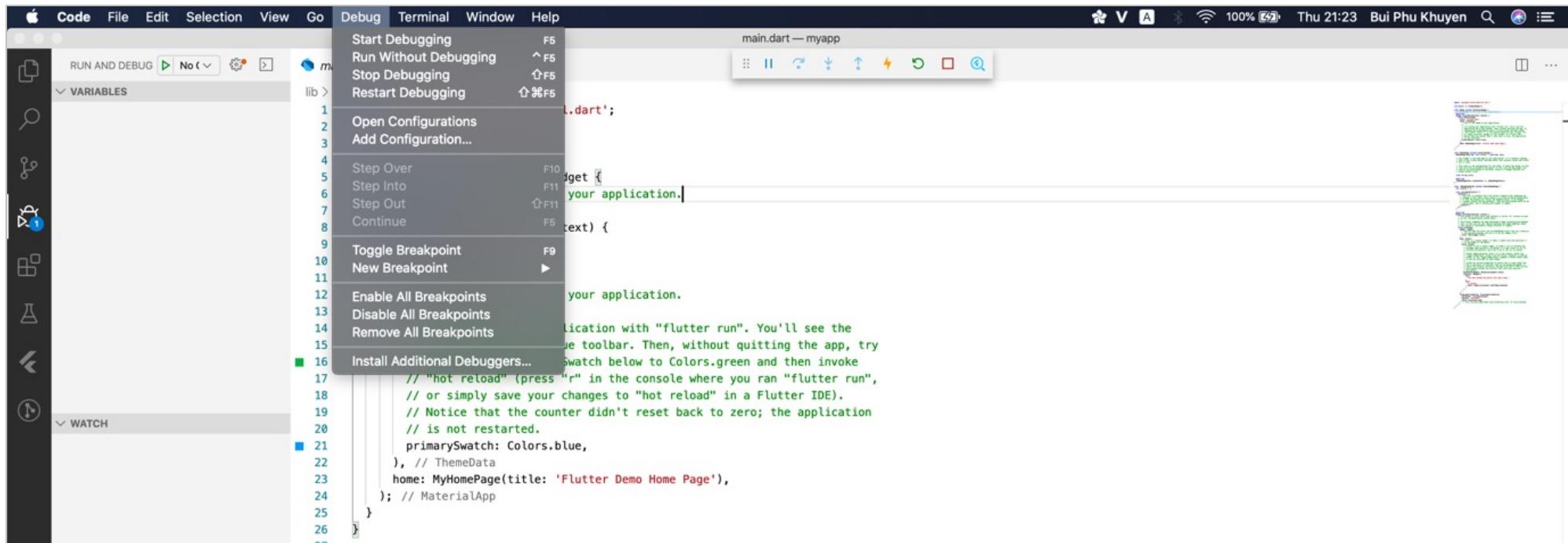
Sẽ chạy dự án

Thông tin phiên bản Dart & Flutter

Ln 1, Col 1 Spaces: 2 UTF-8 LF Dart Flutter: 1.12.13+hotfix.8 Phu Khuyen's iPhone (ios) ⌂ ⓘ

☐ Chạy dự án

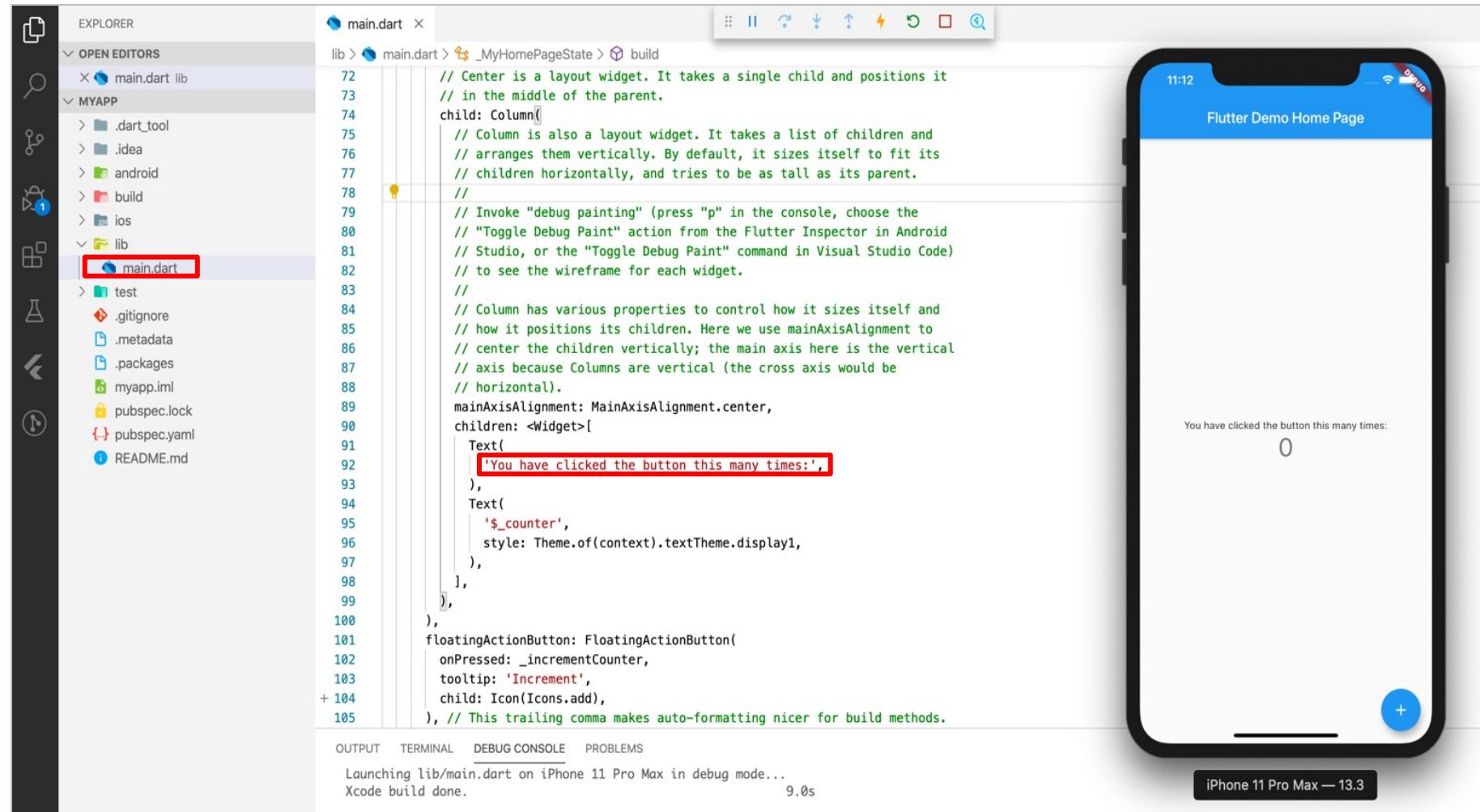
- Chọn Debug > Start Debugging
- Hoặc lệnh: flutter run



The screenshot shows the Android Studio interface with the 'Debug' menu open. The 'Start Debugging' option is highlighted. The code editor displays a Dart file named 'main.dart' with some basic Flutter application code. The bottom right corner of the slide shows the date and time as 'Thu 21:23'.

```
1 lib>
2   L.dart';
3
4   Open Configurations
5   Add Configuration...
6
7   Step Over F10
8   Step Into F11
9   Step Out F11
10 Continue F5
11
12 Toggle Breakpoint F9
13 New Breakpoint ▶
14
15 Enable All Breakpoints
16 Disable All Breakpoints
17 Remove All Breakpoints
18 Install Additional Debuggers...
19
20 // "hot reload" (press "r" in the console where you ran "flutter run",
21 // or simply save your changes to "hot reload" in a Flutter IDE).
22 // Notice that the counter didn't reset back to zero; the application
23 // is not restarted.
24 primarySwatch: Colors.blue,
25 ),
26 // ThemeData
27 home: MyHomePage(title: 'Flutter Demo Home Page'),
28 );
29 // MaterialApp
```

TÍNH NĂNG HOT RELOAD



Cho phép thấy ngay lập tức các thay đổi trong mã nguồn mà không cần khởi động lại ứng dụng.

CẤU TRÚC DỰ ÁN FLUTTER

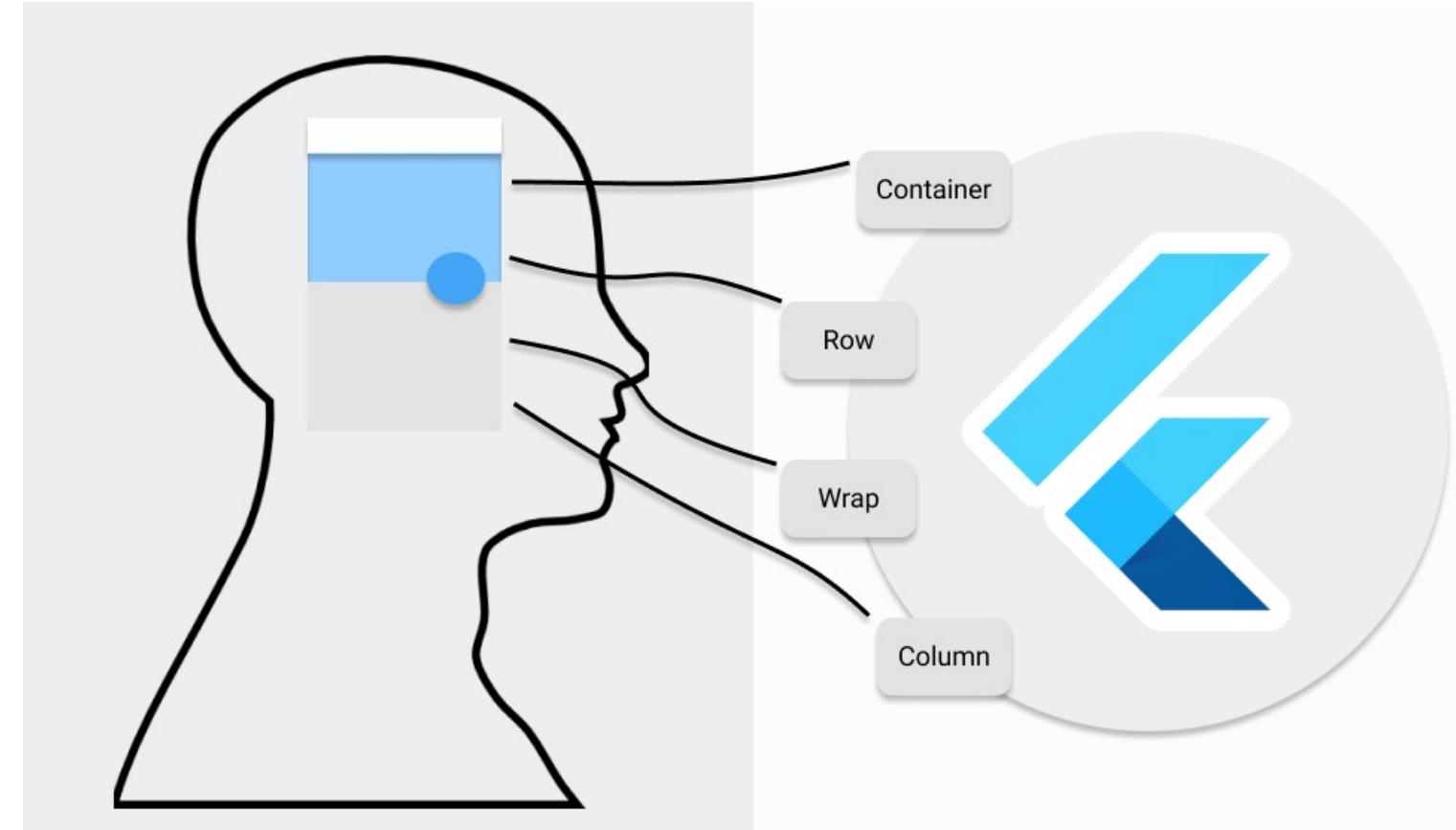
- **pubspec.yaml**: File cấu hình chính của dự án, chứa thông tin **dependencies** và **asset**.
- **lib/**: Thư mục chứa mã nguồn dự án (có chứa file **main.dart**).
- **android/**, **ios/**, **web/**, **windows/...**: Mã nguồn và cấu hình cho từng nền tảng.
- **build/**: Thư mục tạo tự động khi build dự án.

```
> .dart_tool
> .idea
> android
> build
> ios
> lib
> linux
> macos
> test
> web
> windows
> .gitignore
> .metadata
> analysis_options.yaml
> flutter_application_demo.iml
> pubspec.lock
> pubspec.yaml
> README.md
```



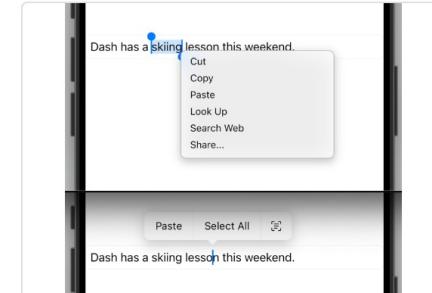
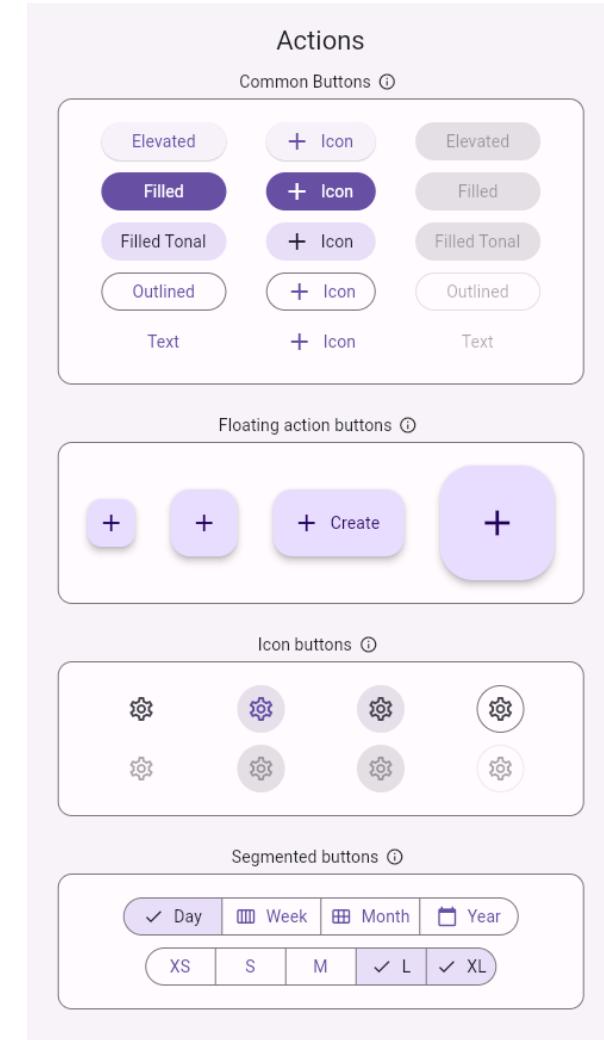
TÌM HIỂU CƠ BẢN VỀ FLUTTER

- **Widget** là các **thành phần cơ bản** để xây dựng giao diện trong Flutter.
- Mọi thứ trong Flutter đều là một **Widget**: văn bản, nút bấm, bố cục...



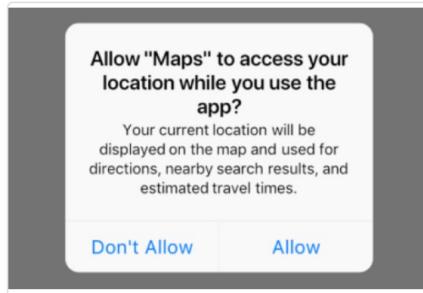
Phân loại Widgets

- Material Widgets: Dành cho Android, thiết kế theo Material Design của Google.
- Cupertino Widgets: Dành cho iOS, thiết kế theo Human Interface Guidelines của Apple.



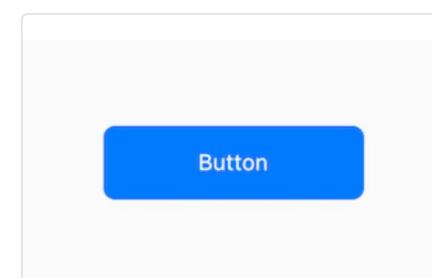
CupertinoAdaptiveTextSelectionToolbar

The default Cupertino context menu for text selection for the current platform with the given children.



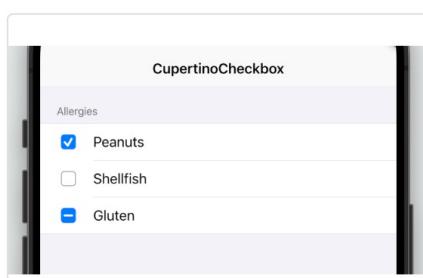
CupertinoAlertDialog

An iOS-style alert dialog.



CupertinoButton

An iOS-style button.

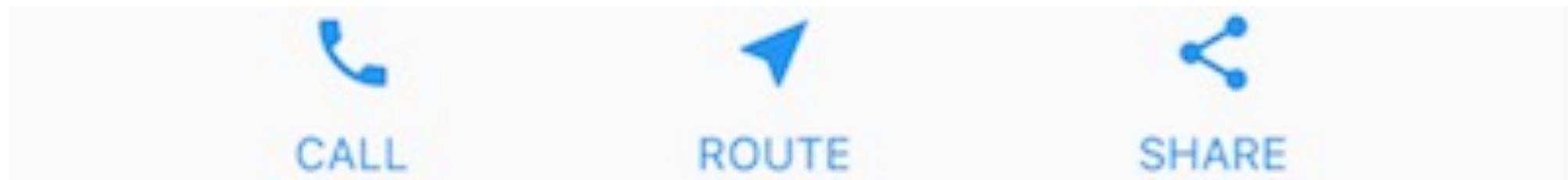


CupertinoCheckBox

A macOS-style checkbox.

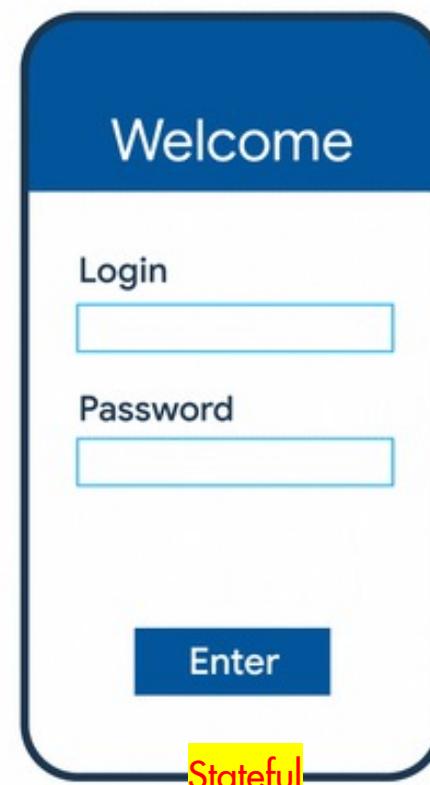
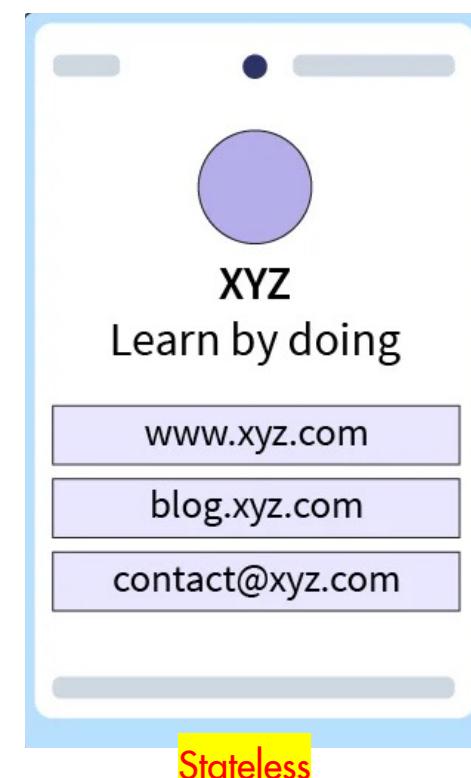
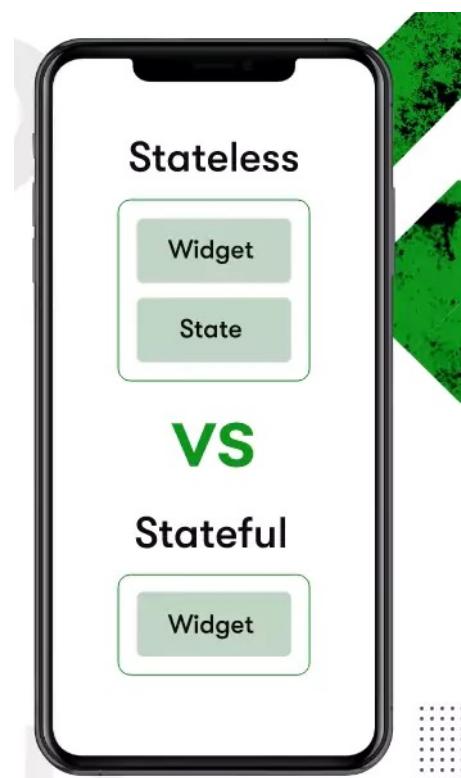
WIDGETS LAYOUT

Widgets Layout là các lớp dùng để xây dựng giao diện người dùng (UI) trong Flutter. Chúng không chỉ được sử dụng để tạo các thành phần giao diện mà còn để bố trí các thành phần này. Bằng cách kết hợp các widget đơn giản, ta có thể xây dựng nên các widget phức tạp hơn



STATELESS & STATEFUL WIDGETS

- **StatelessWidget:** Không có trạng thái, chỉ hiển thị dữ liệu cố định.
- **StatefulWidget:** Có thể thay đổi trạng thái và cập nhật giao diện khi cần.



- **Local State Management:** Quản lý trạng thái cục bộ trong widget. (Sử dụng setState())
- **Global State Management:** Quản lý trạng thái toàn cục, chia sẻ giữa nhiều widget hoặc toàn bộ ứng dụng. (Sử dụng các thư viện: Provider, Get, Bloc)

```
int _counter = 0;  
  
void _incrementCounter() {  
    setState(() {  
        _counter++;  
    });  
}
```

```
floatingActionButton: FloatingActionButton(  
    onPressed: _incrementCounter,  
    tooltip: 'Increment',  
    child: Icon(Icons.add),  
,
```

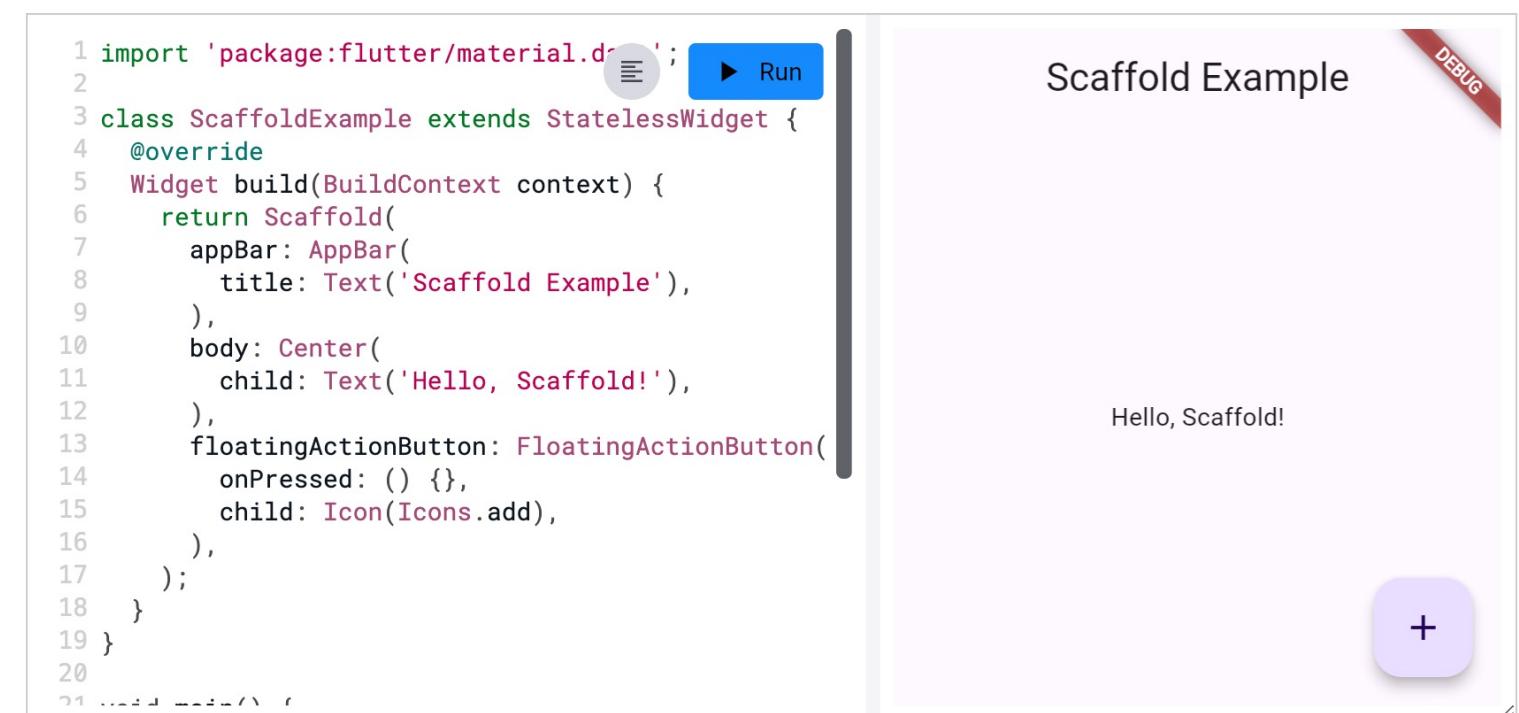
Ví dụ minh họa sử dụng setState() để quản lý trạng thái cục bộ



TÌM HIỂU CƠ BẢN WIDGETS PHỔ BIẾN

Scaffold là widget cơ bản trong Material Design cung cấp cấu trúc giao diện cho các trang màn hình ứng dụng

```
Scaffold({  
  Key? key,  
  this.appBar,  
  this.body,  
  this.floatingActionButton,  
  // Các thuộc tính khác  
})
```



The screenshot shows the code for a Scaffold example in the Android Studio code editor. The code defines a StatelessWidget named ScaffoldExample that returns a Scaffold with an AppBar titled 'Scaffold Example', a Centered body with the text 'Hello, Scaffold!', and a floating action button. A 'Run' button is visible above the code. To the right, the emulator displays the resulting app with the text 'Hello, Scaffold!' and a floating action button. A red ribbon at the top right of the emulator screen says 'DEBUG'. A purple '+' button is located in the bottom right corner of the emulator window.

```
1 import 'package:flutter/material.dart'; ► Run  
2  
3 class ScaffoldExample extends StatelessWidget {  
4   @override  
5   Widget build(BuildContext context) {  
6     return Scaffold(  
7       appBar: AppBar(  
8         title: Text('Scaffold Example'),  
9       ),  
10      body: Center(  
11        child: Text('Hello, Scaffold!'),  
12      ),  
13      floatingActionButton: FloatingActionButton(  
14        onPressed: () {},  
15        child: Icon(Icons.add),  
16      ),  
17    );  
18  }  
19 }  
20  
21
```

Scaffold Example

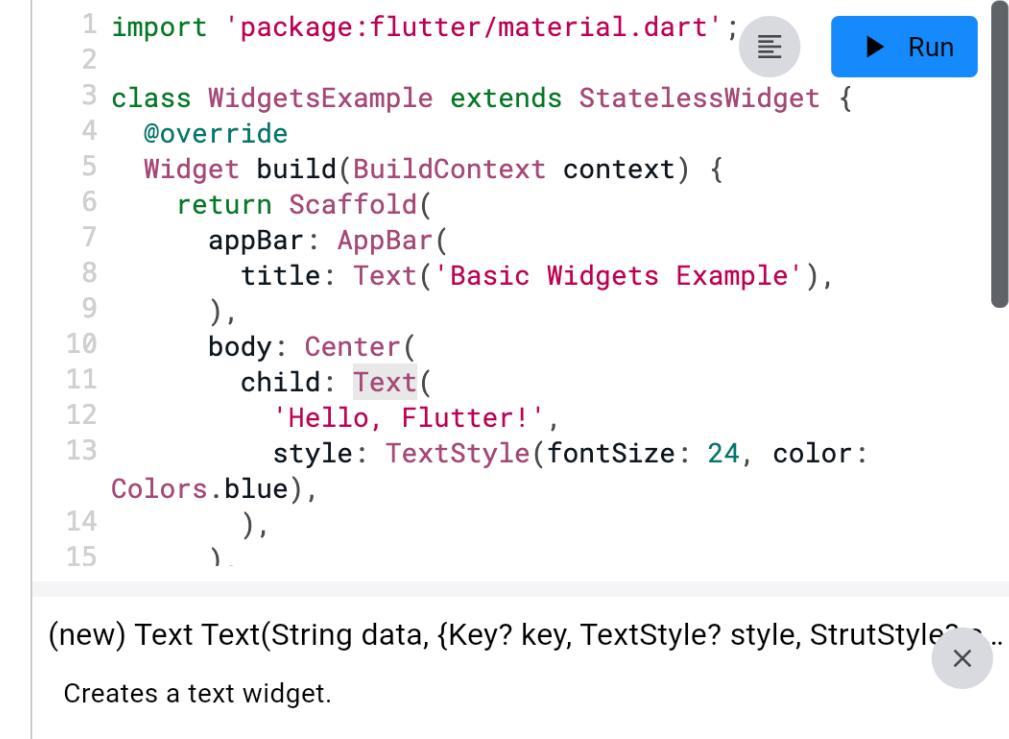
Hello, Scaffold!

DEBUG

+

Text hiển thị một đoạn văn bản đơn giản

```
Text(  
    String data, {  
        Key? key,  
        this.style,  
        this.textAlign,  
        this.overflow,  
        // Các thuộc tính khác  
    })
```



```
1 import 'package:flutter/material.dart';  
2  
3 class WidgetsExample extends StatelessWidget {  
4     @override  
5     Widget build(BuildContext context) {  
6         return Scaffold(  
7             appBar: AppBar(  
8                 title: Text('Basic Widgets Example'),  
9             ),  
10            body: Center(  
11                child: Text(  
12                    'Hello, Flutter!',  
13                    style: TextStyle(fontSize: 24, color:  
14                        Colors.blue),  
15                ),  
16            ),  
17        );  
18    }  
19}
```

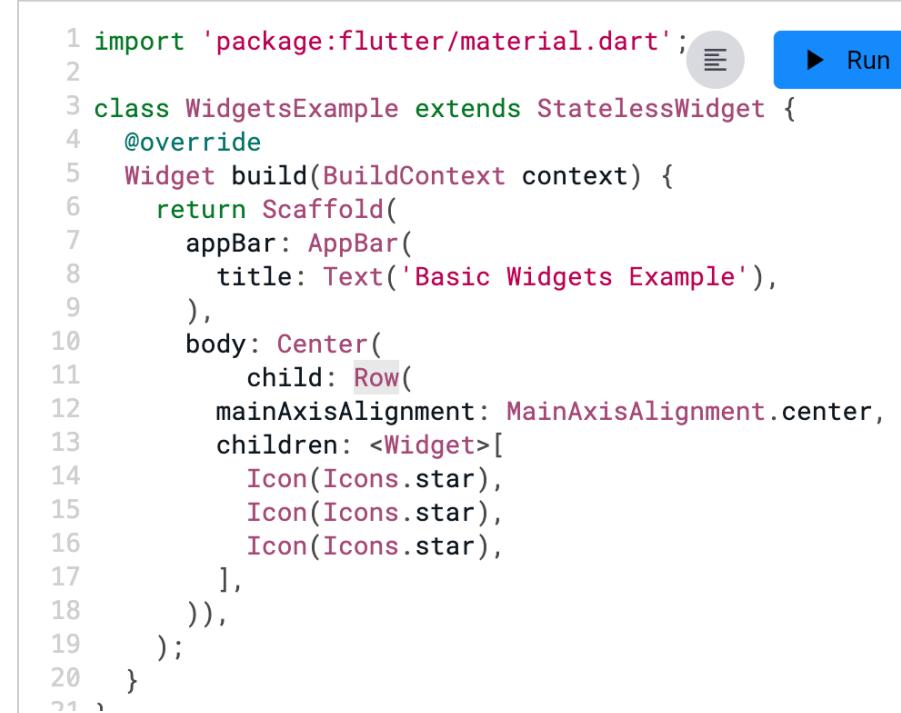
(new) Text Text(String data, {Key? key, TextStyle? style, StrutStyle? style, TextOverflow? overflow})
Creates a text widget.

Basic Widgets Example

Hello, Flutter!

Row sắp xếp các widget con theo chiều ngang.

```
Row({  
  Key? key,  
  this.mainAxisAlignment =  
  MainAxisAlignment.start,  
  this.crossAxisAlignment =  
  CrossAxisAlignment.center,  
  this.children = const <Widget>[],  
  // Các thuộc tính khác  
})
```



```
1 import 'package:flutter/material.dart';  
2  
3 class WidgetsExample extends StatelessWidget {  
4   @override  
5   Widget build(BuildContext context) {  
6     return Scaffold(  
7       appBar: AppBar(  
8         title: Text('Basic Widgets Example'),  
9       ),  
10      body: Center(  
11        child: Row(  
12          mainAxisAlignment: MainAxisAlignment.center,  
13          children: <Widget>[  
14            Icon(Icons.star),  
15            Icon(Icons.star),  
16            Icon(Icons.star),  
17          ],  
18        )),  
19      );  
20    }  
21  }
```

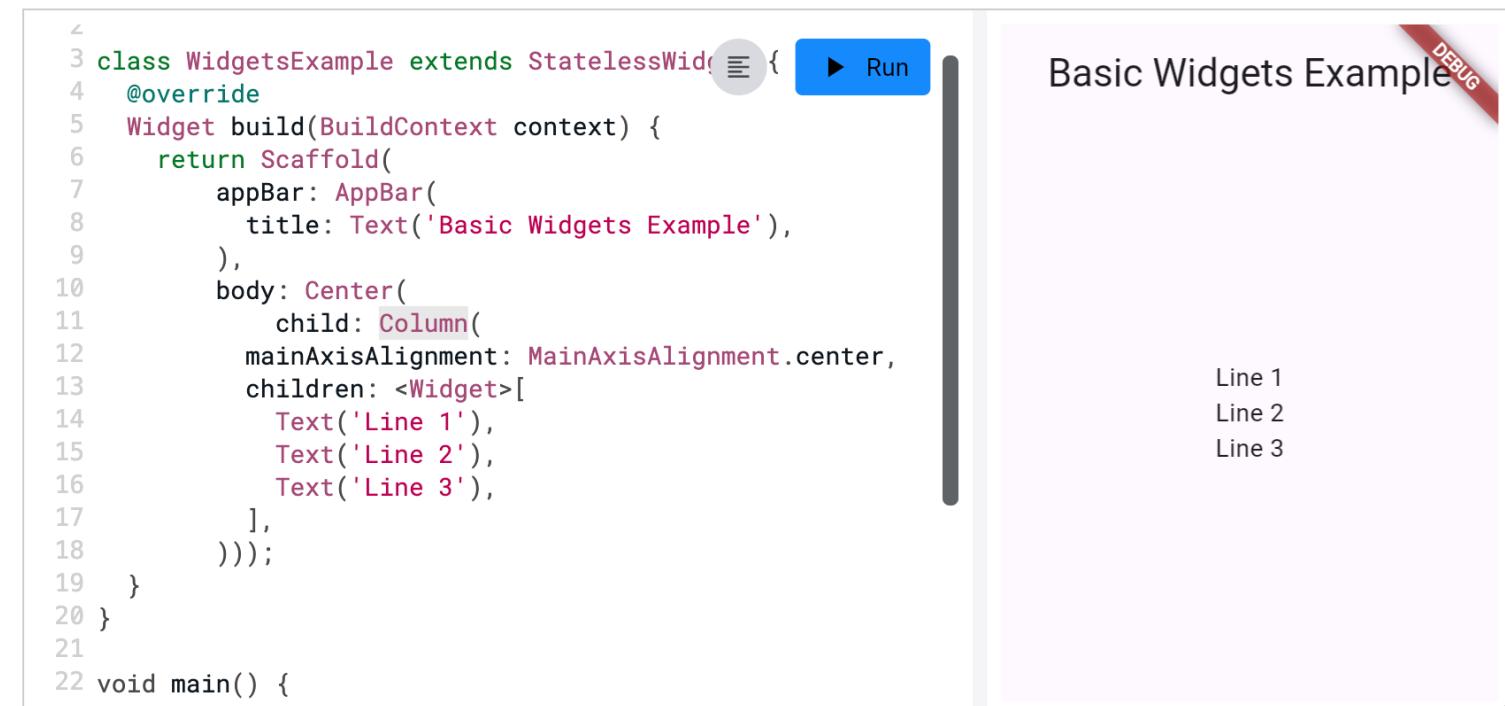
Basic Widgets Example

DEBUG



Column sắp xếp các widget con theo chiều dọc.

```
Column({  
    key: key,  
    mainAxisAlignment: MainAxisSize.start,  
    crossAxisAlignment: CrossAxisAlignment.center,  
    children: const <Widget>[],  
    // Các thuộc tính khác  
})
```



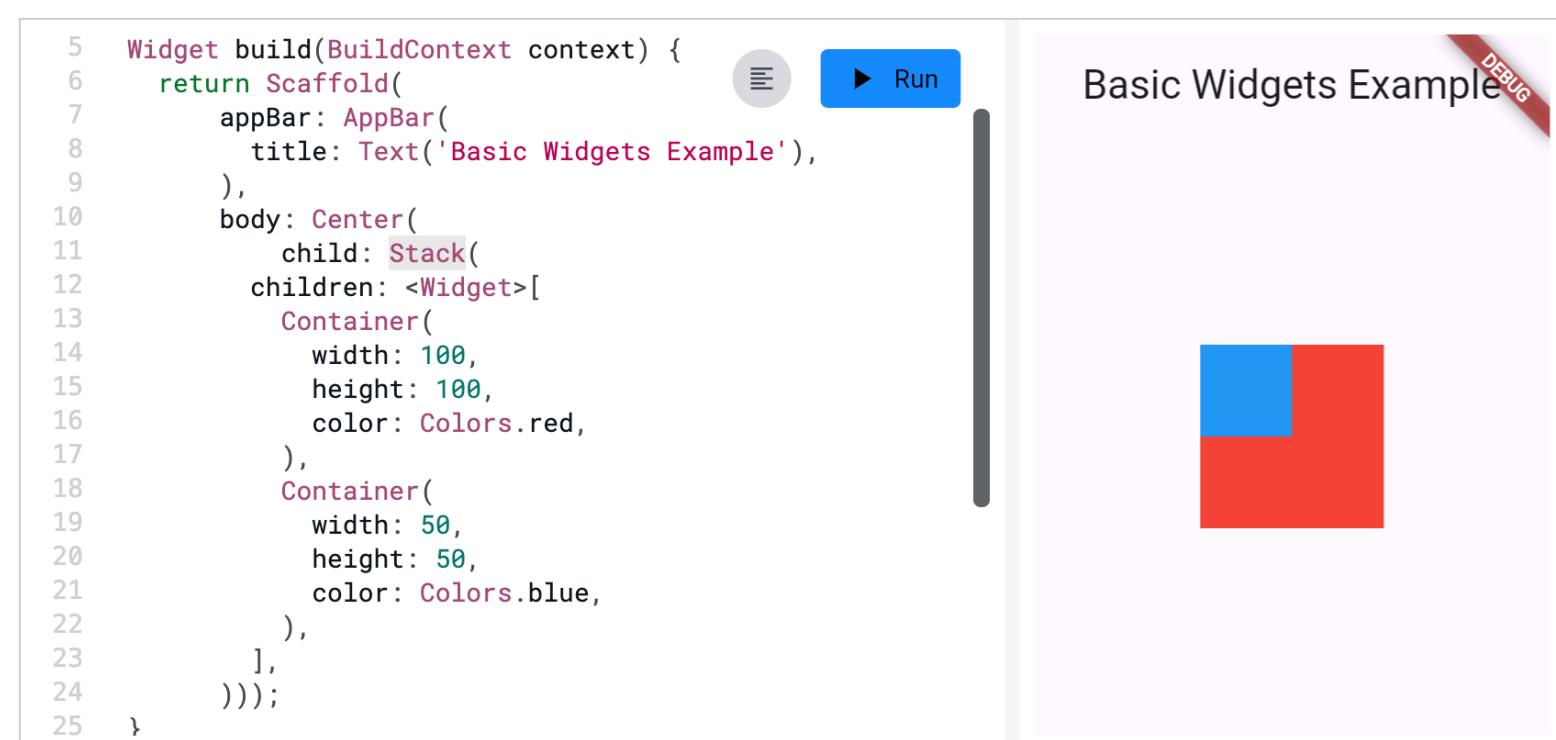
Basic Widgets Example

```
 3 class WidgetsExample extends StatelessWidget {  
 4   @override  
 5   Widget build(BuildContext context) {  
 6     return Scaffold(  
 7       appBar: AppBar(  
 8         title: Text('Basic Widgets Example'),  
 9       ),  
10      body: Center(  
11        child: Column(  
12          mainAxisAlignment: MainAxisAlignment.center,  
13          children: <Widget>[  
14            Text('Line 1'),  
15            Text('Line 2'),  
16            Text('Line 3'),  
17          ],  
18        )),  
19      );  
20    }  
21    void main() {  
22    }  
23  }  
24  
```

Line 1
Line 2
Line 3

Stack xếp chồng các widget con lên nhau.

```
Stack({  
  Key? key,  
  this.alignment =  
  AlignmentDirectional.topStart,  
  this.fit = StackFit.loose,  
  this.children = const <Widget>[],  
  // Các thuộc tính khác  
})
```



Basic Widgets Example

```
5 Widget build(BuildContext context) {  
6   return Scaffold(  
7     appBar: AppBar(  
8       title: Text('Basic Widgets Example'),  
9     ),  
10    body: Center(  
11      child: Stack(  
12        children: <Widget>[  
13          Container(  
14            width: 100,  
15            height: 100,  
16            color: Colors.red,  
17          ),  
18          Container(  
19            width: 50,  
20            height: 50,  
21            color: Colors.blue,  
22          ),  
23        ],  
24      ));  
25}
```

Container là một widget linh hoạt để chứa các widget con với khả năng áp dụng các thuộc tính như padding, margin, border và màu nền.

```
Container({  
  Key? key,  
  this.alignment,  
  this.padding,  
  this.color,  
  this.decoration,  
  this.child,  
  // Các thuộc tính khác  
})
```

```
1 import 'package:flutter/material.dart';  
2  
3 class WidgetsExample extends StatelessWidget {  
4   @override  
5   Widget build(BuildContext context) {  
6     return Scaffold(  
7       appBar: AppBar(  
8         title: Text('Basic Widgets Example'),  
9       ),  
10      body: Center(  
11        child: Container(  
12          padding: EdgeInsets.all(16.0),  
13          color: Colors.blue,  
14          child: Text('Hello, Container!'),  
15        )),  
16      )  
17    }  
18  
19 void main() {  
20   runApp(  
21     // Your app's root widget here  
22   );  
23 }
```



Run

Basic Widgets Example

Hello, Container!

Align căn chỉnh widget con theo một vị trí cụ thể.

```
Align({  
  Key? key,  
  this.alignment = Alignment.center,  
  this.child,  
  // Các thuộc tính khác  
})
```

The screenshot shows the Flutter code editor interface. On the left, the code for `WidgetsExample` is displayed, including imports, class definition, build method, and main function. The `Align` widget in the `body` is highlighted with a pink rectangle. On the right, the resulting application is shown with the title "Basic Widgets Example". A red banner at the top right of the screen says "DEBUG". The text "Top Right" is displayed in the center of the application's body.

```
1 import 'package:flutter/material.dart';
2
3 class WidgetsExample extends StatelessWidget {
4   @override
5   Widget build(BuildContext context) {
6     return Scaffold(
7       appBar: AppBar(
8         title: Text('Basic Widgets Example'),
9       ),
10      body: Center(
11        child: Align(
12          alignment: Alignment.topRight,
13          child: Text('Top Right'),
14        )));
15    }
16  }
17
18 void main() {
19   runApp(
20     MaterialApp(
21       title: 'Flutter Demo' // used by the OS task switcher
22     ),
23   );
24 }
```

Padding thêm khoảng cách xung quanh widget con.

```
Padding({  
  Key? key,  
  this.padding,  
  this.child,  
})
```

```
1 import 'package:flutter/material.dart';  
2  
3 class WidgetsExample extends StatelessWidget {  
4   @override  
5   Widget build(BuildContext context) {  
6     return Scaffold(  
7       appBar: AppBar(  
8         title: Text('Basic Widgets Example'),  
9       ),  
10      body: Center(  
11        child: Container(  
12          color: Colors.blueAccent,  
13          child: Padding(  
14            padding: EdgeInsets.all(16.0),  
15            child: Text('Padded Text', style:  
16              TextStyle(color: Colors.white)),  
17            ),  
18            ),  
19          );  
20    }  
}
```



Basic Widgets Example

DEBUG

Padded Text

SizedBox đặt kích thước cụ thể cho widget con hoặc tạo khoảng trống giữa các widget.

```
SizedBox({  
  Key? key,  
  this.width,  
  this.height,  
  this.child,  
})
```

```
5   Widget build(BuildContext context) {  
6     return Scaffold(  
7       appBar: AppBar(  
8         title: Text('Basic Widgets Example'),  
9       ),  
10      body: Center(  
11        child: Column(  
12          children: [  
13            Text('Love', style: TextStyle(fontSize: 24)),  
14            SizedBox(height: 20.0),  
15            Text('Like', style: TextStyle(fontSize: 24)),  
16          ]));  
17      }  
18    }  
19  
20 void main() {  
21   runApp(  
22     MaterialApp(  
23       title: 'My app', // used by the OS task switcher  
24       home: SafeArea(  
25         ...  
26       ),  
27     ),  
28   );  
29 }
```

Basic Widgets Example

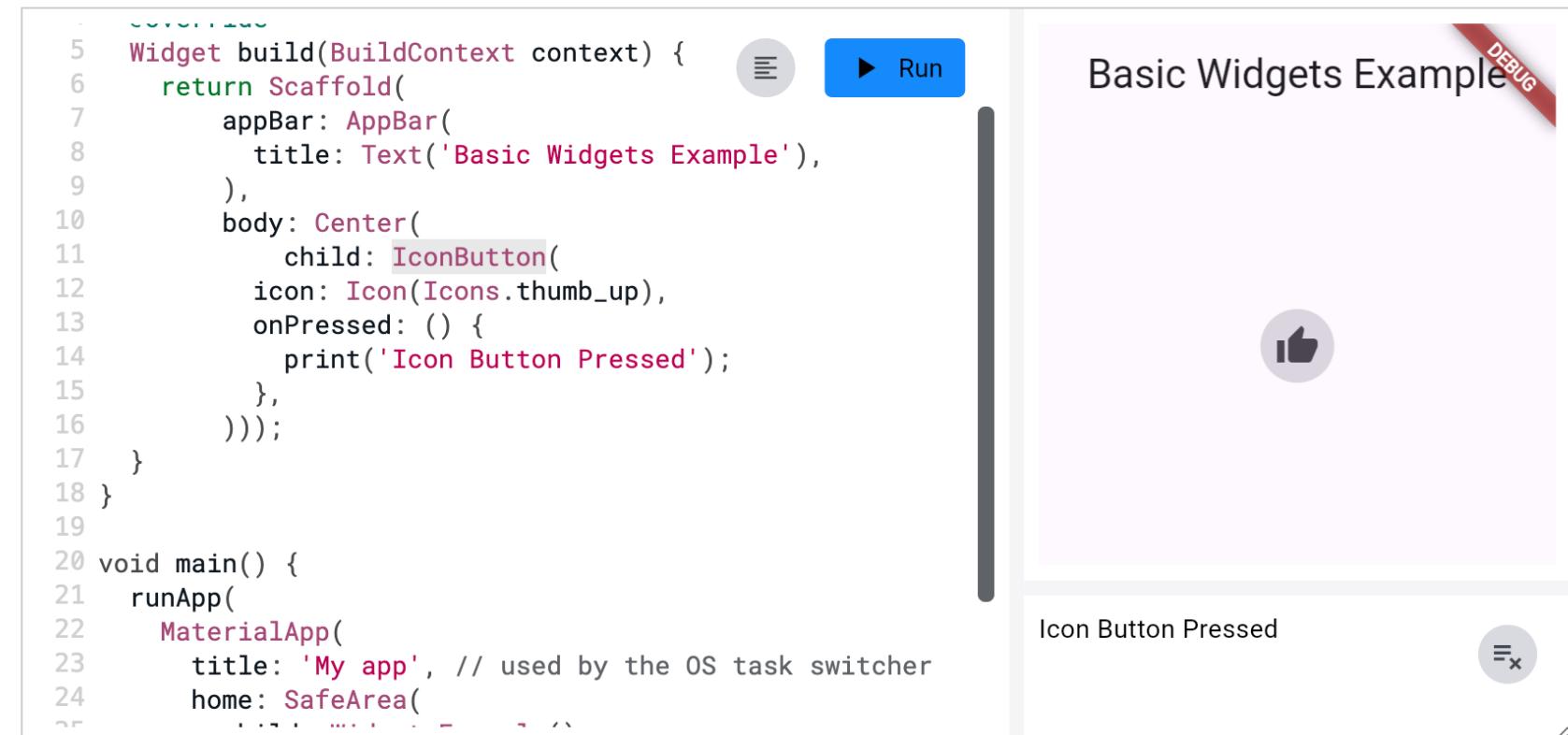
DEBUG

Love

Like

IconButton là nút bấm hiển thị biểu tượng, có thể bấm được.

```
IconButton({  
    Key? key,  
    this.icon,  
    this.onPressed,  
    // Các thuộc tính khác  
})
```



```
class Services {  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(  
        title: Text('Basic Widgets Example'),  
      ),  
      body: Center(  
        child: IconButton(  
          icon: Icon(Icons.thumb_up),  
          onPressed: () {  
            print('Icon Button Pressed');  
          },  
        ),  
      ),  
    );  
  }  
}  
  
void main() {  
  runApp(  
    MaterialApp(  
      title: 'My app', // used by the OS task switcher  
      home: SafeArea(  
        ...  
      ),  
    ),  
  );  
}
```



TÌM HIỂU CƠ BẢN ASSET VÀ MEDIA

- Trong Flutter, **assets** là các tập tin được đóng gói và triển khai cùng ứng dụng, có thể truy cập tại **runtime**.
- Các loại **assets** phổ biến bao gồm dữ liệu tĩnh (ví dụ: tệp JSON), tệp cấu hình, biểu tượng và hình ảnh (JPEG, PNG, GIF, WebP, BMP).
- Sử dụng tệp **pubspec.yaml** để khai báo các **assets** cần thiết.

flutter:

assets:

- `assets/my_icon.png`
- `assets/background.jpeg`
- `assets/data/config.json`

Xử lý và hiển thị Assets:

- Sử dụng **AssetImage** để tải hình ảnh của widget.

```
return Image(image: AssetImage('assets/background.png'));
```

- Sử dụng **rootBundle** từ `package:flutter/services.dart` để tải các tệp văn bản hoặc JSON

```
import 'package:flutter/services.dart' show rootBundle;  
  
Future<String> loadAsset() async {  
  
  return await rootBundle.loadString('assets/data/config.json');  
}
```

XỬ LÝ ASSETS VÀ IMAGES {3}

Ngoài ra, ta có thể tải hình ảnh từ Internet bằng cách sử dụng **Image.network**:

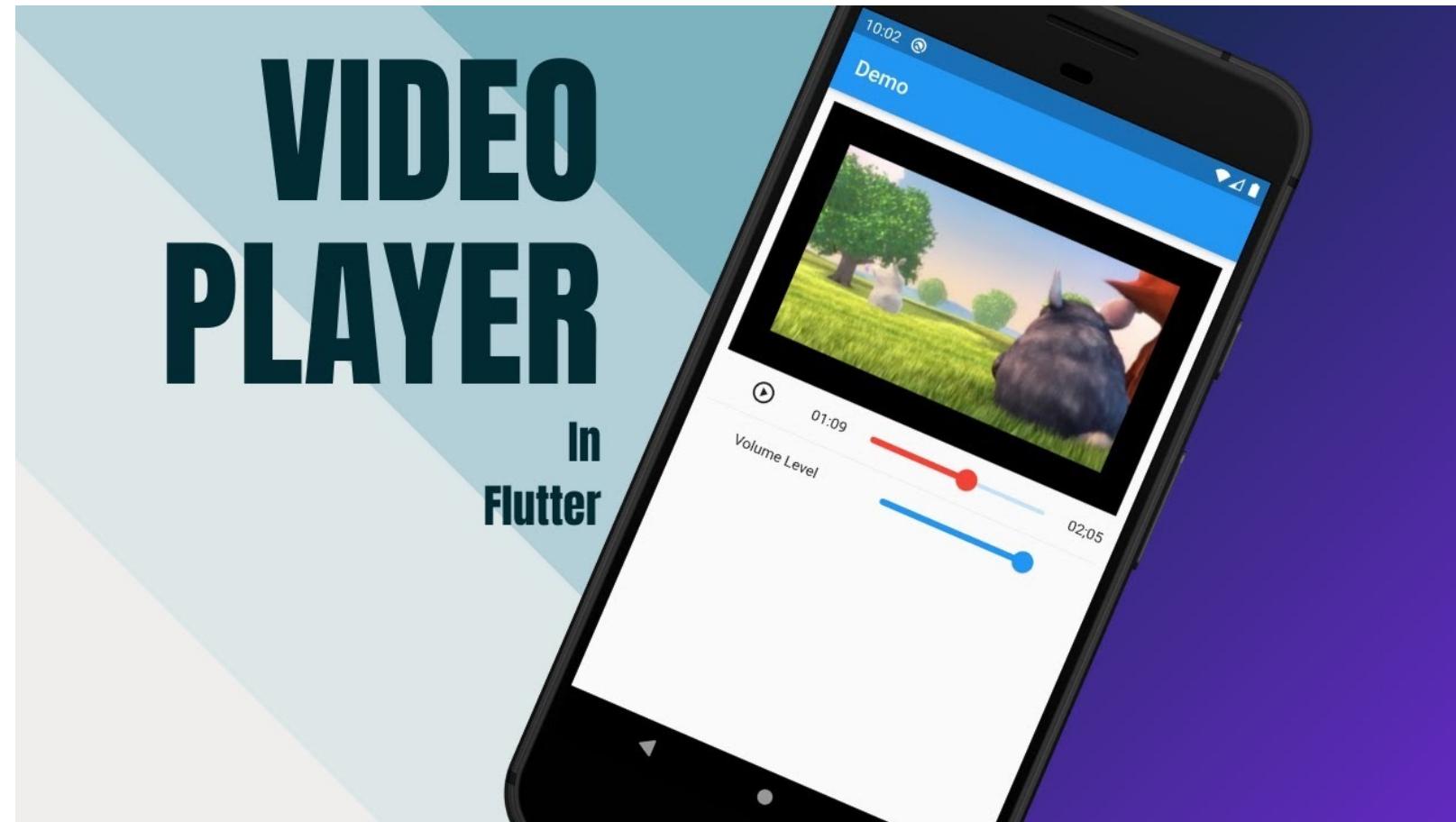


Image.network('<https://flutter.dev/simple.png>')

<https://flutter.dev/simple.png>

XỬ LÝ ASSETS VÀ IMAGES {4}

Sử dụng plugin **video_player** để phát video từ hệ thống tệp hoặc từ internet.





TÌM HIỂU CƠ BẢN

NAVIGATING & ROUTING

- **Navigator** là một widget quản lý các màn hình như một ngăn xếp.
- Mỗi khi chúng ta chuyển sang một màn hình mới, nó sẽ được đẩy vào ngăn xếp và khi quay lại, màn hình sẽ được bật ra khỏi ngăn xếp.
- Để điều hướng đến 1 màn hình mới, ta sử dụng phương thức **push()** và để quay lại, ta sử dụng **pop()**.

```
// Within the First Route widget
 onPressed: () {
 Navigator.push(
 context,
 MaterialPageRoute(builder: (context)
 => const SecondRoute()),
 );
}
```

```
// Within the SecondRoute widget
 onPressed: () {
 Navigator.pop(context);
}
```

<https://docs.flutter.dev/cookbook/navigation/navigation-basics>

- Named routes cho phép ta định nghĩa tất cả các tuyến đường trong ứng dụng tại một chỗ duy nhất, giúp mã nguồn trở nên gọn gàng và dễ quản lý hơn

```
MaterialApp(  
    title: 'Named Routes Demo',  
    initialRoute: '/',  
    routes: {  
        '/': (context)  
            => const FirstScreen(),  
        '/second': (context)  
            => const SecondScreen(),  
    },  
)
```

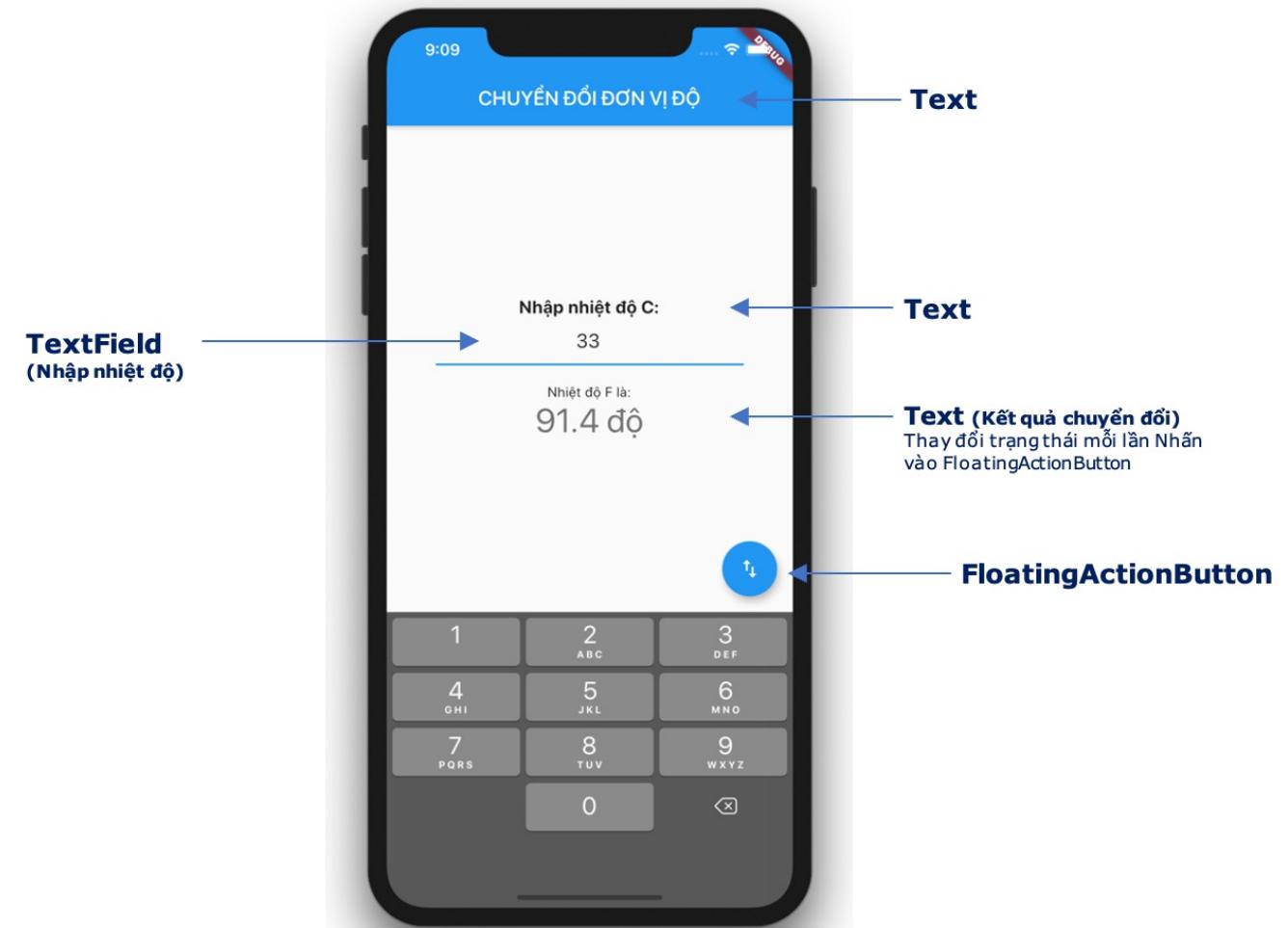
```
// Within the `FirstScreen` widget  
onPressed: () {  
    Navigator.pushNamed(context, '/second');  
}
```

```
// Within the SecondScreen widget  
onPressed: () {  
    Navigator.pop(context);  
}
```

<https://docs.flutter.dev/cookbook/navigation/named-routes>

BÀI TẬP VẬN DỤNG

Dựa trên dự án Demo mẫu của Flutter, hãy xử lý lại main.dart với chức năng chuyển đổi đơn vị độ (từ độ C sang độ F)



Cảm ơn

Các bạn đã chú ý lắng nghe



HỎI ĐÁP



Q&A